Mobile and Pervasive Computing 23/24

Project Draft – BrightBalance+

Authors: 64783 André Correia

64813 Rodrigo Fontinha

68759 Tiago Martinho

April 11th, 2024

Description

The idea behind this project originated from the human necessity to have a conscious and respectful attitude towards the use of everyday resources, which has become more prevalent over the years. The environments we know today are littered with devices capturing and storing more and more information in digital format, and sometimes that becomes excessive and expensive to maintain.

BrightBalance+ is an iteration of a previous project – BrightBalance – that never came to real fruition. It consists of a smart system that focuses on light management in an attempt to automate light emission and intensity in a given household. The idea is to take the presence or absence of natural sunlight to, respectively, turn the lights off or on, raise or lower window blinds and so on.

Although the main purpose of the system is to manage lighting, BrightBalance+ seeks to monitor and react to temperature as well.

Architecture

The architecture constitutes a **N** clients to **1** service relationship and splits the system into three different components:

- Mobile application – Allows the communication between users and the service;
- Ubiquitous application – The combination of a microcontroller with sensors, actuators and a program, specialized to sensing the environment for phenomena that trigger context-aware actions;
- Webserver – Database server to store and provide convenient access to system and user data.

Figure 1 – System Architecture

The webserver of choice, Firebase, will serve as the central point of BrightBalance+, storing user and sensor data, and communicating with both the mobile and ubiquitous applications when needed.

Mobile Application

The BrightBalance+ mobile application will be implemented with Android Studio, using the Flutter framework.

In terms of functionality, users will be allowed to create their own profile to interact with the service. System specific user preferences, or “Moods” as designated in the application, to be stored in the webserver, will define properties like custom light colors and preferred light emission intensity to allow users to apply their own twist to the service. Moods will facilitate management across different rooms, as it will also be possible for users to create a representation of the divisions of their house in the application. Speaking of rooms, each will include an on-demand system on or off toggle, and display information about temperature and the amount of people currently in it.

Additionally, notifications will be sent to the user whenever the ubiquitous component is ready to take action regarding the raising and lowering of blinds and turning lights on and off, ultimately providing control to the user over BrightBalance+.

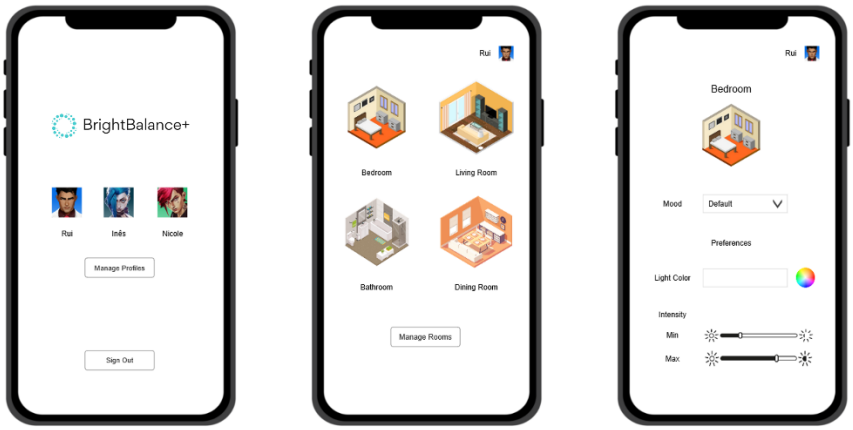

Figure 2 – BrightBalance+ Mockup

Ubiquitous Application

BrightBalance+ will require a microcontroller to interact with the environment of the user’s household. The module of choice will be the same ESP32 (see Fig. 3) used initially for BrightBalance.

Figure 3 – WiFi module, Espressif ESP32 S2 WROOM

The following Table 1 includes the sensors required to allow the correct behavior of the system, particularly in collecting information about its surrounding environments.

Sensor	Role	Image
2x LDR (Light Dependent Resistor) Photoresistors	Comparing house interior luminosity against exterior daylight	
2x NTC (Negative Temperature Coefficient) Thermistors	Comparing interior with exterior temperature	

Table 1 – BrightBalance+ Sensors

The necessary actuators can be found in Table 2.

Actuator	Role	Image
RGB LED (Light Emitter Diode)	Light emission based on environment conditions and active Mood	
Blinds	Room ambience based on sensor readings	
Fan (or other air conditioner system)	Room conditioning in case a room is too hot	 

Table 2 – BrightBalance+ Actuators

Instead of using an actual fan and blinds, the ubiquitous system will use a green and a red LED to simulate the fan being turned on and off, respectively, and a servo motor to simulate the act of raising and lowering of the blinds.